UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/761,105	01/20/2004	Kamrul Alam	END920030142US1	3124

37945 7590 03/13/2008

DUKE W. YEE
YEE AND ASSOCIATES, P.C.
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
RICHARDSON, THOMAS W	
ART UNIT	PAPER NUMBER
2144	

MAIL DATE	DELIVERY MODE
03/13/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/761,105	ALAM ET AL.	
	Examiner	Art Unit	
	THOMAS RICHARDSON	2144	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 December 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 December 2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-15 are pending for examination.

Claims 1-15 are rejected.

Drawings

The drawings were received on 20 December, 2007. These drawings are accepted, and the objection to the drawings is therefore lifted.

Response to Amendment

The examiner lifts previous rejection of claims 1-15 in view of applicant arguments. Reference Johnson fails under U.S.C. 103(c), and the previous rejection is therefore overcome.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

3. Claims 1, 2, 4-6, 8, 10-12, and 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over WO 03/005192, Ossiansson, et al in combination with US 6 697 795, Holcomb.

4. As per claim 1, Ossiansson teaches a method for accessing a first application by a first server and then replacing said first application with a second application executing in a second server, said first server having a first, local storage, said second server having a second, local storage (page 2, lines 16-25), said method comprising the steps of:

routing a request for an application to said first server, said first server requesting from said first local storage said application, said request being redirected from said first local storage to said first application in a shared storage, wherein the file name of said first application forms a hierarchical directory in said shared storage (page 3, lines 5-15, where the first server can access the operating system (OS1) in the shared storage means, also page 4, lines 13-21, where the version of the program is applicable to OS1); and

subsequently routing a request for said application to a second server, said second server requesting from said second local storage, said request being redirected from said second local storage to said second application in said shared storage, said second application in said shared storage having a name different than said name of said first application in said shared storage, wherein said name of said second application forms a hierarchical directory in said shared storage (page 3, lines 5-15, where the second server can access the other operating system (OS2) in the shared

storage means, also page 4, lines 13-21, where another version of the application is applicable to OS2).

Ossiansson does not teach a file system to implement, but Holcomb discloses a method for creating a virtual file system, wherein the system forms a hierarchic data structure:

> said first application in said shared storage having said first level name, said second level name and a third level name, wherein said first level name, said second level name and said third level name of said first application form a hierarchical directory in said shared storage (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory); and
>
> said second application in said shared storage having said first level name, said second level name and a third level name different than said third level name of said first application in said shared storage, wherein said first level name, said second level name and said third level name of said second application form a hierarchical directory in said shared storage (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory. The differing files, e.g. F85 and F86, show that two files may contain the same high-level names, with the difference in name only on the level of the actual file).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ Holcomb's file naming system in Ossiansson's operating system server. Holcomb states that the system allows for large storage while using little memory

(Column 1, lines 16-20). This would be beneficial in Ossiansson's system, as it would allow the server to store more information in an easily accessible manner.

5. As per claim 2, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 1 wherein said second application is a more recent version of said first application (Ossiansson teaches this limitation. It is well known in the art that versioning can be used to differentiate between newer and older files, and versions are used to differentiate files in shared storage as shown on page 4, lines 13-15).

6. As per claim 4, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 1 further comprising a third server which accesses from said shared storage said first application identified by said first level name, said second level name and said third level name of said first application prior to the subsequently routing request step (Ossiansson teaches this limitation. Page 5, lines 16-18, where the load balance unit routes between users and servers).

7. As per claim 5, Ossiansson teaches a method for accessing a first application by a first server and then replacing said first application with a second application executing in a second server, said first server having a first, local storage, said second server having a second, local storage (page 2, lines 16-25), said method comprising the steps of:

routing a request for an application identified by a first hierarchical directory from a proxy server to said first server, said first server requesting from said first local storage said application identified by said first hierarchical directory, said first server request being redirected from said first local storage to said first

application in said shared storage (page 3, lines 5-15, where the first server can access the operating system (OS1) in the shared storage means, also page 4, lines 13-21, where the version of the program is applicable to OS1), and subsequently routing a request for said application identified by said first hierarchical directory to a second server, said second server requesting from said second local storage said application identified by said first hierarchical directory, said second server request being redirected from said second local storage to said second application in said shared storage (page 3, lines 5-15, where the second server can access the other operating system (OS2) in the shared storage means, also page 4, lines 13-21, where another version of the application is applicable to OS2).

Ossiansson does not teach a file system to implement, but Holcomb discloses a method for creating a virtual file system, wherein the system forms a hierarchic data structure

said first application in said shared storage having a second, extended hierarchical directory comprising said first hierarchical directory plus a lower level qualifier (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory); and

said second application in said shared storage having a third, extended hierarchical directory comprising said first hierarchical directory plus a lower level qualifier different than that of said second, extended hierarchical directory (Figure 1 and accompanying description, where the files F85-F88 are shown at the

bottom of a hierarchical data structure given by a multi-tiered directory. The differing files, e.g. F85 and F86, show that two files may contain the same high-level names, with the difference in name only on the level of the actual file).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ Holcomb's file naming system in Ossiansson's operating system server. Holcomb states that the system allows for large storage while using little memory (Column 1, lines 16-20). This would be beneficial in Ossiansson's system, as it would allow the server to store more information in an easily accessible manner.

8. As per claim 6, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 5 wherein said second application is a more recent version of said first application (Ossiansson teaches this limitation. It is well known in the art that versioning can be used to differentiate between newer and older files, and versions are used to differentiate files in shared storage as shown on page 4, lines 13-15).

9. As per claim 8, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 5 further comprising a third server which accesses from said shared storage said first application identified by said second, extended hierarchical directory prior to the subsequently routing request step (Ossiansson teaches this limitation. Page 5, lines 16-18, where the load balance unit routes between users and servers).

10. As per claim 10, Ossiansson teaches a method for controlling access to first and second applications in a shared storage (page 2, lines 16-25), said method comprising the steps of:

a first server requesting from a first local storage a copy of an application identified by a first level name and a second level name, said first server request being redirected from said first local storage to said first application in said shared storage (page 3, lines 5-15, where the first server can access the operating system (OS1) in the shared storage means, also page 4, lines 13-21, where the version of the program is applicable to OS1); and

subsequently, a second server requesting from a second local storage a copy of an application identified by said first level name and said second level name, said second server request being redirected from said second local storage to said second application in said shared storage (page 3, lines 5-15, where the second server can access the other operating system (OS2) in the shared storage means, also page 4, lines 13-21, where another version of the application is applicable to OS2).

Ossiansson does not teach a file system to implement, but Holcomb discloses a method for creating a virtual file system, wherein the system forms a hierarchic data structure

said first application in said shared storage having said first level name, said second level name and a third level name, wherein said first level name, said second level name and said third level name of said first application form a hierarchical directory in said shared storage (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory); and

said second application in said shared storage having said first level name, said second level name and a third level name different than said third level name of said first application in said shared storage, wherein said first level name, said second level name and said third level name of said second application form a hierarchical directory in said shared storage (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory. The differing files, e.g. F85 and F86, show that two files may contain the same high-level names, with the difference in name only on the level of the actual file).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ Holcomb's file naming system in Ossiansson's operating system server. Holcomb states that the system allows for large storage while using little memory (Column 1, lines 16-20). This would be beneficial in Ossiansson's system, as it would allow the server to store more information in an easily accessible manner.

11. As per claim 11, Ossiansson teaches a method for controlling access to first and second applications in a shared storage (page 2, lines 16-25), said method comprising the steps of:

a first server requesting from a first local storage a copy of an application identified by a first hierarchical directory, said first server request being redirected from said first local storage to said first application in said shared storage (page 3, lines 5-15, where the first server can access the operating system (OS1) in the

shared storage means, also page 4, lines 13-21, where the version of the program is applicable to OS1), and

subsequently, a second server requesting from a second local storage a copy of an application identified by said first hierarchical directory, said second server request being redirected from said second local storage to said second application in said shared storage (page 3, lines 5-15, where the second server can access the other operating system (OS2) in the shared storage means, also page 4, lines 13-21, where another version of the application is applicable to OS2).

Ossiansson does not teach a file system to implement, but Holcomb discloses a method for creating a virtual file system, wherein the system forms a hierarchic data structure

said first application in said shared storage having a second, extended hierarchical directory comprising said first hierarchical directory plus a lower level qualifier (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory); and

said second application in said shared storage having a third, extended hierarchical directory comprising said first hierarchical directory plus a lower level qualifier different than the lower level qualifier of said second, extended hierarchical directory (Figure 1 and accompanying description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory. The differing files, e.g. F85 and F86, show that two files

may contain the same high-level names, with the difference in name only on the level of the actual file).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ Holcomb's file naming system in Ossiansson's operating system server. Holcomb states that the system allows for large storage while using little memory (Column 1, lines 16-20). This would be beneficial in Ossiansson's system, as it would allow the server to store more information in an easily accessible manner.

12. As per claim 12, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 11 wherein said second application is a more recent version of said first application (Ossiansson teaches this limitation. It is well known in the art that versioning can be used to differentiate between newer and older files, and versions are used to differentiate files in shared storage as shown on page 4, lines 13-15).

13. As per claim 14, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 11 further comprising a third server which accesses from said shared storage said first application identified by said second, extended hierarchical directory prior to the subsequently routing request step (Ossiansson teaches this limitation. Page 5, lines 16-18, where the load balance unit routes between users and servers).

14. As per claim 15, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 11 wherein said first hierarchical directory comprises at least two levels of qualifiers (Holcomb teaches this limitation. Figure 1 and accompanying

description, where the files F85-F88 are shown at the bottom of a hierarchical data structure given by a multi-tiered directory).

15. Claims 3, 7, 9, and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ossiansson and Holcomb as applied to claims 1, 5, and 11 above, and further in view of US 7 206 852, Ferguson et al.

16. As per claim 3, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 1.

The combination does not teach rerouting an application request. Ferguson teaches an application update system wherein there is a proxy server which routed said request for said application identified by said first level name and said second level name to said first server, and further comprising the step of reconfiguring said proxy server to route subsequent requests for said application identified by said first level name and said second level name to said second server instead of said first server (Figure 1 and accompanying descriptions show that in prior art, an application server may be removed from the system to be upgraded, meanwhile the requests are sent to another server).

It would have been obvious to one of ordinary skill in the art at the time of invention to use Ferguson's disclosed method of rerouting server requests in the system of Ossiansson. The system would benefit, as it would allow traffic to be immediately rerouted to another server if the application was upgraded. After a file has been modified, the older version is no longer used, so it would be unnecessary to route requests to it. The changing of routing allows the host to have continuous access, which improves server availability and system capacity, which is beneficial and often essential

(Ferguson, Column 1, lines 27-31). In addition, the inclusion of Holcomb's file naming system would also have been obvious to one of ordinary skill in the art at the time of the invention. Holcomb states that the system allows for large storage while using little memory (Column 1, lines 16-20).

17. As per claim 7, the combination of Johnson and Holcomb teaches a method as set forth in claim 5.

The combination does not teach rerouting an application request. Ferguson teaches an application update system wherein there is a proxy server which routed said request for said application identified by said first hierarchical directory to said first server, and further comprising the step of reconfiguring said proxy server to route subsequent requests for said application identified by said first hierarchical directory to said second server instead of said first server (Figure 1 and accompanying descriptions show that in prior art, an application server may be removed from the system to be upgraded, meanwhile the requests are sent to another server).

It would have been obvious to one of ordinary skill in the art at the time of invention to use Ferguson's disclosed method of rerouting server requests in the system of Ossiansson. The system would benefit, as it would allow traffic to be immediately rerouted to another server if the application was upgraded. After a file has been modified, the older version is no longer used, so it would be unnecessary to route requests to it. The changing of routing allows the host to have continuous access, which improves server availability and system capacity, which is beneficial and often essential (Ferguson, Column 1, lines 27-31). In addition, the inclusion of Holcomb's file naming

system would also have been obvious to one of ordinary skill in the art at the time of the invention. Holcomb states that the system allows for large storage while using little memory (Column 1, lines 16-20).

18. As per claim 13, the combination of Ossiansson and Holcomb teaches a method as set forth in claim 11.

The combination does not teach rerouting an application request. Ferguson teaches an application update system wherein there is a proxy server which routed said request for said application identified by said first hierarchical directory to said first server, and further comprising the step of reconfiguring said proxy server to route subsequent requests for said application identified by said first hierarchical directory to said second server instead of said first server (Figure 1 and accompanying descriptions show that in prior art, an application server may be removed from the system to be upgraded, meanwhile the requests are sent to another server).

It would have been obvious to one of ordinary skill in the art at the time of invention to use Ferguson's disclosed method of rerouting server requests in the system of Ossiansson. The system would benefit, as it would allow traffic to be immediately rerouted to another server if the application was upgraded. After a file has been modified, the older version is no longer used, so it would be unnecessary to route requests to it. The changing of routing allows the host to have continuous access, which improves server availability and system capacity, which is beneficial and often essential (Ferguson, Column 1, lines 27-31). In addition, the inclusion of Holcomb's file naming system would also have been obvious to one of ordinary skill in the art at the time of the

invention. Holcomb states that the system allows for large storage while using little memory (Column 1, lines 16-20).

Conclusion

19. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US 6 912 571, Serena teaches a method of replacing content.

US 7 103 617, Phatak teaches a method and system for use in storage caching with a distributed file system.

US 6 360 366 and US 6 006 034, Heath et al teach a system and method for version upgrading and maintenance.

US 5 403 639, Belsan et al teaches a file server having snapshots of applications.

US 5 706 510, Burgoon teaches a symbolic link history management system.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to THOMAS RICHARDSON whose telephone number is (571) 270-1191. The examiner can normally be reached on Monday through Thursday, 8am-5pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Vaughn can be reached on (571) 272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TR
2/25/2008

/William C. Vaughn, Jr./
Supervisory Patent Examiner, Art Unit 2144